

雲端運算軟體即服務診所資訊系統永續運作模式之研究

陳啟維
雲科大資管所研究生
g9823234@yuntech.edu.tw

孫培然
中山附醫資訊室主任
peiran.sun@gmail.com

黃錦法
雲科大資管所副教授
huangcf@mis.yuntech.edu.tw

摘要

在健保總額固定之下，醫療院所紛紛尋找開源節流解決方案，尤其是開業醫療診所，希望在不增加軟硬體及人事開銷，又可以依照診所需求選擇資訊服務，再依服務的多寡計費；就如使用同自來水、電力、鐵路等以量計費的方式。如今在雲端運算(Cloud Computing)其一應用模式，軟體即服務 (Software as a Services ,SaaS)，便可實現以量計費及隨插即用(Plug And Play)之願景。

SaaS 服務架構，雖然可以成為診所降低資訊營運成本的解決方案，但依附在公眾網路下的 SaaS 服務，一旦網路不穩定或斷線時，其服務便無法持續運作。若要確保網路非常穩定及永不中斷嚴苛條件下，必須花費許多網路備援設備，其成本相當昂貴，且不符合成本效益，導致醫療診所不敢貿然嘗試。

為了實現以量計費及隨插即用的理想，以及確保醫療資訊服務品質能夠永續服務，本研究係採用 Sync Framework 同步技術，搭配本研究提出的資料變動同步矩陣，建置一個網路中斷時，還能夠永續運作之 SaaS 服務模式。解決了網路不穩定或斷線，造成醫療資訊服務中斷之缺陷。此外，本研究遵循服務導向架構 (Service Oriented Architecture ,SOA) 架構，預留服務無限擴充能力，能夠快速滿足用戶因系統整合，衍生額外資料存取之需求。使用本研究所提出的永續運作 SaaS 服務，讓醫療診所享受 SaaS 服務模式帶來的各種優點，降低營運資訊成本，專注醫療本業，進而提昇醫療服務品質。

關鍵詞：雲端運算、軟體即服務、同步技術、服務導向架構。

1. 緒論

在健保總額固定之下，醫療院所紛紛尋找開源節流的解決方案。不單單只是國內醫療產業，受近年金融海嘯之累，全球各行各業莫不以減少支出，降低營運成本最優先考量目標。市場調查研究公司 (Forrester Research, Inc)，在 2009 年的研究顯示，全球企業約有八成的資訊費用是花費在硬體維修支出上，另外已有四成以上的企業開始縮減在資訊費用的支出 (Forrester, 2009)。在資訊費用大幅減少狀況下，又要維持企業內部 IT 正常運作；甚至背負必須加速企業 e 化以降低營運成本嚴苛使命下，雲端運算似乎已成為企業資訊部門，解決問題的最佳選擇方案之一。根據 Gartner、Forrester、MIC 等研究組織分析，普遍都認為企業導入雲端運算之後可減少硬體建置費用支出、降低 IT 人力成本、以及大幅縮短軟體導入與建置時程。

雲端運算中的軟體即服務(Software as a Service ,SaaS)應用模式，讓企業在不購置新的硬體設備及僱用額外軟體開發軟體人員狀況下，只要透過網路進行資料存取設定後，定期支付服務供應商授權費，獲得軟體使用授權後，即可使用企業所需的軟體服務。軟體使用過程中產生的資料，將固定儲存在資訊服務提供者端。爾後軟體所需的伺服器運作營運成本、資料備份、軟體維護及更新等 IT 費用支出，將由 SaaS 服務提供廠商概括承擔全部費用。

但有專家學者提到，如果無法連接到網際網路，雲端運算就完全行不通了(miller, 2008)。在目前尚無網路提供者能提供 7 x24 小時，網路不中斷的服務品質情況下，倘若為了讓雲端運算服務正常運作，因而建置多方網路備援機制將不敷成本。但企業在營運上又不允許因網路中斷後導致資訊

服務暫停狀況發生；如何享受雲端運算帶來的便利性，又可以避免因網路中斷導致資訊服務暫停的狀況發生是目前雲端運算待解決的問題之一。

研究是以欲使用雲端運算 SaaS 服務之醫療診所為案例，擬歸納目前業界目前常用離線資料處理應用方法，提出雲端運算 SaaS 服務應用模式，在網路中斷時醫療資訊系統能夠繼續提供服務之研究，透過研究希望可以達到下面三個目的：

1. 提出 SaaS 服務永續運作模式之建議。
2. 提出資料同步模式最佳化之資料變動同步策略矩陣建議。
3. 所獲得研究成果可以提供醫療診所或資訊業界運用與參考。

2. 文獻探討

2.1 雲端運算

「電腦運算能力有一天可能會變成諸如自來水、電力、鐵路之類隨插即用的公用設施」(McCarthy, 1961)，在四十餘年後這個想法終於靠雲端運算實現了。所謂的「雲端運算」(Cloud Computing)就是泛指所有的網路運用。雲端運算的關鍵就是「雲端」，也就是大規模的伺服器網路，或者甚至是相互連接成網路的個人電腦，這些電腦平行地運作，結合各電腦的資源，產生有如超級電腦的運算能力 (miller, 2008)。

從另一角度來看雲端；「雲端」可以分成「雲服務」與「端裝置」兩個面向。所謂的「雲服務」，係將平台或硬體(泛指 IT 基礎設備、虛擬化機器或儲存設備)包裝成一種服務，而「端裝置」則是提供多用戶利用且多元運用的裝置，並連結與存取「雲服務」所提供的服務功能。「雲服務」與「端裝置」應用及功能呈現方式如圖 1 所示。



圖 1 雲服務與端裝置
資料來源：(王耀聰, 2010)

根據美國國家標準局(NIST)對雲端運算所下的定義，雲端運算必須具備下列特性：

1. 三種服務架構(Service Models):
基礎架構服務 (Infrastructure as a Services, IaaS)、平台服務(Platform as a Services, PaaS)、軟體即服務(Software as a Services, SaaS)。如圖 2 所示。



圖 2 雲端三種服務架構
資料來源：(gipi, 2009)

2. 四種佈署模型(Deployment Models):
公有雲(Public Cloud)、私有雲(Private Cloud)、社群雲(Community Cloud)、混合雲(Hybrid Cloud)。
3. 五種基礎特徵(Essential Characteristics):
隨選自助服務 (On-Demand Self-Service)、隨時隨地用任何網路裝置存取(Broad Network Access)、多人共享資源池(Resource Pooling)、快速重新佈署靈活度 (Rapid Elasticity)、服務可被量測與監控(Measured Service) (Peter Mell and Tim Grance, 2010)。

2.2 SaaS 軟體即服務

軟體是需要持續的服務、維護和升級，才能正常發揮其內在價值(葉偉等, 2009)。

雲端運算其中一項服務架構，SaaS 服務的精神就是把軟體需求當成一項服務；也就是服務提供商把企業對軟體需求，轉換成一種服務提供方式。

傳統軟體取得方式是，企業買斷軟體使用授權，有些軟體尚需要額外簽訂軟體維護合約狀況下，在軟體產品有限的使用周期內，始獲得開發商的軟體諮詢、軟體問題修正及軟體版本更新…等服務。在 SaaS 服務模式中，企業可依據企業需求獲得所需軟體服務，除支付與服務供應商約定使用授權費外，就無需再支付額其他費用了；換言之，SaaS 服務可降低軟體授權費用 (Anthony et al., 2010)。

SaaS 服務可免除自行開發所需的軟體週期，讓組織快速獲得所需應用程式 (Anthony et al., 2010)。同時，供應商可以依照企業需求隨時動態調整所需軟體服務。當企業軟體服務需求持續時，只要繼續支付與供應商與企業約定使用授權費，就可以合法使用軟體服務，及持續獲得軟體產品使用周期間的各項服務，當企業軟體服務需求不復存在，並停止支付使用授權費時，即宣告停止使用該項軟體服務。

傳統軟體取得方式，企業必須概括承擔軟體導入建置失敗的風險，以及導入軟體成效與評估預期不合之損失。SaaS 服務模式，全程透過 Internet 公眾網路連線；進行軟體的導入建置、授權及最後的上線使用。倘若軟體服務無法成功導入及建置，或是上線後，軟體服務使用不如預期成效時，企業可立即中止使用該軟體服務，用多少就付多少；因此在 SaaS 服務模式下，企業必須承擔的風險及損失是相當有限。

但在現實中，SaaS 服務並非完美無瑕，仍有缺點及待克服問題，如下所列：

1. 完全依附在網路架構下，有"無法永續"的隱憂 (Drdo et al., 2009)：

SaaS 服務全程是透過網路運作，用戶沒有了網路就無法獲得服務，換句話說 SaaS 服務的效能，完全被網路品質左右。目前尚無網路提供者保證能提供 7 x24 網路不中斷品質服務，若配置多方網路備援機制，確保網路不中斷

所耗費的成本，是非常昂貴的；有些企業用戶在營運中，是完全無法容忍網路中斷即停止服務風險，解決方案正是本研究的目的之一。

2. 資料的完整、可用、機密性議題：
文獻指出 SaaS 服務，儲存在供應商的資料可能不安全 (Miller, 2008)、有資料與身分的安全風險 (Charles, 2010) 以及資料保密性和資訊安全的隱憂 (Drdo et al., 2009)。SaaS 服務運作過程中，所產出的資料完全儲存在服務提供者硬體設備上，用戶很難直接取用資料或對資料做備份動作。資料的完整、可用性對企業資訊部門是最重視的議題之一，由於 SaaS 服務也不過是近年剛形成的新興服務，對於廠商提供的保證，用戶還是會擔憂；因為資料一旦出錯或損毀時，用戶將束手無策。唯一解決之道，只有靠用戶選擇服務提供者時，慎選能提供高可靠度資料儲存環境之服務提供者。在資料機密性部分：再怎麼完善的資安政策，還是有可能出現漏洞。無論資料存放在企業內部，或是存放在服務提供者端，都有可能因人為因素而造成資料外洩。企業內部應該以更嚴謹的態度制定導入雲端運算 (SaaS 軟體即服務) 政策或原則，將機密性的隱憂降為最低。
3. 無法與公司既有軟體整合隱憂 (Drdo et al., 2009)：

目前並無 SaaS 服務統一的標準或規範，而且服務運作產生存放在提供者的資料，企業不易隨意取得再利用。服務提供者本身有自己的資料處理模式，在無客製化的特殊需求下，是很難與企業本身已運行之系統做整合。另一方面，企業在使用一家以上 SaaS 軟體即服務供應商時 (例如：承租 A 供應商銷售系統 (POS) 服務，承租 B 供應商客戶關係管理 (CRM) 服務，承租 C 供應商…等使用狀況)，欲將供應商之間，不同平台產出的資料，做資料橫向交換處理時，是有困難性的。如須

與服務供應商訂定客製化服務需求，來解決類似整合議題時，軟體導入成本勢必暴增。為了解決資料整合議題，相信不久將來應該會制定 SaaS 軟體即服務統一的標準或規範。

2.3 Sync Framework 同步框架技術

近年來企業營運模式擴展，以及行動裝置(筆記型電腦、掌上電腦(PDA)或智慧型手機)資訊產品的日漸普及。員工在辦公室以外使用行動裝置擷取資訊，必須擷取到與在辦公室內一致的資訊，是企業營運上極重要的基本需求(Edmund, 2010)。

透過 Internet 公眾網路經過認證後連接企業網路，擷取最新資訊是目前最普遍的做法，但是礙於目前網路基礎建設仍未十分普及、完善、網路供應商無法提供網路不中斷服務、及企業無法提供快速充足且穩定的頻寬供遠端操作者使用..等狀況下，偶爾連接的應用程式(Occasionally Connected Application, OCA)是目前另一種的解決方案。OCA 可以讓用戶可以在明確離線、使用低頻寬或高延遲網路，或者連接時斷時續的情況下繼續高效率地工作(肖承勇, 2007)。也因為存取資料的動作分散在用戶端，也可大為降低伺服器的負載(Nick, David, Chris & Michael, 2010)。

為了達到 OCA 運作需求，微軟提出同步框架(Sync Framework)技術。Sync Framework 是一個全面性的同步處理平台，可以處理各種應用程式、服務與裝置的共同作業和離線存取(Microsoft Corporation, 2009)，包含主機、個人 PC 及行動裝置使用上；也可用在不同的儲存設備、資料型態、通訊協定等多重的拓樸架構上；Sync Framework 確實可以降低實作 OCA 的複雜性。圖 3 為同步框架技術核心元件。

但是面臨不穩定或低速的網路連線品質下，若同步過多非必要的資料內容，一定會大幅影響 Sync Framework 的同步傳輸效能，因此本研究提出制定有效的資料同步策略，冀望有效率的同步資料；進而協助 Sync Framework 將同步資料能力最佳

化，最終獲得可靠及可用的緩衝資料，讓 SaaS 服務能夠永續運作。

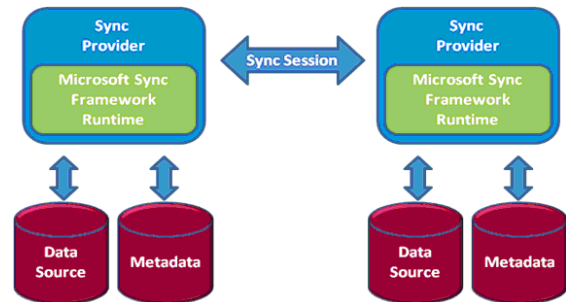


圖 3 Microsoft Synchronization Framework Core Components

資料來源：(Microsoft Corporation, 2010)

2.4 醫療軟體邁入雲端運算概況

在國內，資策會產業情報研究所(MIC)正在建構「醫療雲」，打造「雲端運算醫院資訊系統」，推動制定國家標準化的電子病歷格式；期望病患在國內任何一所醫療機構就診時，利用「醫療雲」服務，即可查到病患歷年所有看診紀錄及報告。對於「醫療雲」的醫療隱私，有關單位應訂立明確法令規範，消除民眾對「醫療雲」侵犯病人隱私的疑慮(MIC, 2010)。

利用雲端運算架構，制定模組，係採用服務導向架構(Service Oriented Architecture, SOA)的概念設計，實作相關軟體元件庫，按需求付費機制，可達到降低軟、硬體相關成本的效果(蔡承宏等, 2010)。另外，可依需取選用臨床資訊服務的 SaaS 平台，是可根據需求建立動態的網路服務組合(蕭惠如等, 2010)，將是未來醫療軟體的趨勢之一。

國內有 45%的醫療機構有配置 11 至 20 專業資訊人員，大約有 30%左右的醫療機構，未能在一天以內將出問題的資訊設備恢復正常(石俊榮, 2010)。資訊設備一旦故障須耗費更多的人力與時間來彌補，因設備故障帶來的不便，嚴重故障時，甚至要中斷醫療診所的醫療服務。雖然，花費高額の資訊成本，但也不能保證資訊設備當日可恢復正常。服務持續、方便、可選擇之醫療軟體服務，就如同水、電等付費

隨開即用的公共服務，是大部分醫療診所美好的願景。

雲端運算的出現，讓醫療診所期待的願景終於有機會實現。SOA 架構是醫療軟體邁入雲端運算重要的基石，軟體服務可以依照 SOA 架構提升成各種雲端運算服務，雲端運算已從願景階段邁入實作及導入階段。但是依附在公眾網路的雲端服務，總是讓醫療診所卻步，沒有網路就等同於設備故障，如何提供離線時永續 SaaS 服務運作，讓醫療診所願意使用雲端計算服務正是本研究的主要目的。

3. 研究架構

雲端運算已邁入實作運行階段，但目前尚無網路提供者，能保證提供 7x24 網路不中斷品質服務。倘若真的要確保網路不中斷，則必須配置多方的網路備援機制，其所耗費網路備援成本將非常的昂貴，導致想導入雲端運算的醫療診所卻步。因此，為了實現以量計費及隨插即用的理想，以及確保醫療服務品質能夠永續服務，本研究係採用 Sync Framework 同步技術，建置一個網路中斷時能夠永續運作之同步議程(Sync Agenda)服務模式架構，著重在使用 SaaS 服務時，在離線(網路斷線)或網路品質不穩定狀況下，服務能夠永續運作，不因離線或網路品質不穩定而中止服務；而在重新連線網路連線品質穩定後，能夠將資料，可靠的同步回存到 SaaS 服務供應商提供遠(雲)端的資料庫主機上後，重新回復原本軟體服務運作方式。圖 4 為 Sync Agenda 的永續服務模式架構元件示意圖。

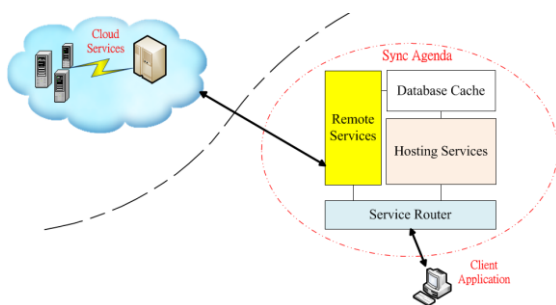


圖 4 永續運作之 SaaS 服務模式架構元件圖

另外針對同步資料需額外善用有限網路頻寬資源下，提出資料同步策略矩陣建議，即使在不穩定的網路品質下，能讓資料同步的效率，提升到最佳狀況，確保同步後的資料可靠且可用，最終達到提供服務永續運作之目的。以下內容將針對本研究提出之 Sync Agenda 的永續服務模式做說明。

本研究架構分為雲端服務 (Cloud Service) 以及在用戶端執行的 Sync Agenda 程式兩個部份，作用如下所述：

1. Cloud Service 部分：

符合 SOA 架構，提供用戶端執行的 Sync Agenda 程式，存取遠(雲)端資料庫資料，其機制類似 Web Services，並作為遠(雲)端資料庫與用戶端緩衝資料庫間資料同步資料時的中介橋樑。

2. 用戶端執行的 Sync Agenda 程式部分：封裝在 Sync Agenda 程式內各元件功能性如下所列：

(1) 服務路由(Service Router)元件：

提供用戶端使用之 SaaS 服務，存取 Cloud Service 提供之遠端資料存取服務，或存取用戶端 Web(Hosting) Service 所提供之用戶端緩衝資料存取服務之路由選擇。在網路正常時提供 Cloud Service 資料存取服務路由，當網路發生異常時，則提供存取用戶端緩衝資料庫的 Web(Hosting) Service 存取服務路由。

(2) 遠端服務(Remote Service)元件：

提供存取 Cloud Service 功能，及負責同步遠(雲)端資料庫與用戶端緩衝資料庫任務，並隨時監測網路狀況，作為提供 Service Router 切換存取路由依據。

(3) 本機網路服務(Hosting Service)元件：

符合 SOA 架構，提供存取用戶端 Database Cache 緩衝資料之 Web(Hosting) Service 功能。

(4) 緩衝資料庫(Database Cache)元件：

存放在用戶端的緩衝資料庫，存放的內容是透過 Remote Service 與遠(雲)端資料庫同步過後的資料，提供用戶

端 Web (Hosting) Service 存取使用之緩衝資料庫。

實際能夠提供永續服務，則是靠 Sync Agenda 程式在用戶端中居中協調運作，其包含的 Service Router、Remote Service、Hosting Service 及 Database Cache 等元件各司其職相互支援運作，期以達到如下目的：

1. 控管資料同步服務：
利用 Sync Framework 同步技術，透過 Remote Services，讓遠(雲)端資料庫與用戶端緩衝資料庫之資料同步化，達到必要資料同步目的。
2. 提供服務存取路由服務：
用戶端使用之 SaaS 軟體即服務應用程式，欲存取資料時，由 Service Router 元件依據網路連線狀況提供服務存取路由。
3. 用戶端資料存取服務：
在網路中斷時，讓 SaaS 服務應用程式，透過用戶端 Web (Hosting) Services 元件存取用戶端緩衝資料庫資料。

另外，確保具有可靠及可用的緩衝資料，是永續運作 SaaS 服務基本要素。除利用 Sync Framework 技術協助同步遠端及用戶端緩衝資料外，在網路頻寬資源有限下，必須要有效同步遠(雲)端及用戶端緩衝資料，且力求同步時間最短化，在下次網路斷線前須完成資料同步動作；有鑑於此嚴苛要求，本研究進一步提出資料變動同步策略矩陣建議，期望將同步效能最佳化。例如：診所營運時，醫師依照患者預約掛號資料順序，依序看診；在診療過程中，醫師詳細診斷病情，並依據病情開出診斷書及藥單，完成診療交易過程。在上述診療過中建議將資料屬性區分：

1. 參考類資料：在第一次啟動 Sync Agenda 程式；第一次執行同步資料作業時，優先同步下載本分類資料，爾後運行 SaaS 服務視需要，再手動執行同步下載資料命令，同步該分類特定資料。例如：藥品資料，只需要在第一次執行 SaaS 服務時，下載同步藥品資料即可，診所營運時，除新藥品上架

外，無須再次同步下載藥品資料。

2. 清單類資料：在每次執行同步資料作業時，必需同時同步上傳及下載本分類資料。例如：醫師看診時，需要下載同步預約掛號清單；在患者完成診療後，需上傳同步預約掛號清單資料。
3. 交易類資料：在每次執行同步資料作業時，只須上傳同步本分類資料。例如：醫師完成交易過程後，同步上載診斷書及藥單交易資料。

Sync Framework 可定義不同資料表，選擇不同資料同步模式，讓資料同步作業具有彈性。首先將需同步之資料，區分資料變動類型後，針對各類型資料，設定最佳同步模式，讓每次執行同步資料作業時，依照此策略進行資料同步，即可達到同步效能最佳化目的，確保緩衝資料的可靠及可用性，讓 SaaS 服務永續。表 1 為本研究建議之資料變動同步策略矩陣。

表 1 資料變動同步策略矩陣

資料變動類型 \ 同步模式	資料上傳	資料下載
參考類資料		V
清單類資料	V	V
交易類資料	V	

4. 研究情境模擬

為了確保在公眾網路不穩定或斷線時還能夠提供永續 SaaS 服務，本研究將針對網路正常連線、網路異常中斷及網路中斷回復正常時的 SaaS 服務流程中，其 Sync Agenda 永續服務模式架構中各元件，是如何分工合作，所採取的應對措施情境模擬如下：

1. 網路正常連線時的 SaaS 服務流程：
Remote Service 會定時依照資料變動同步矩陣策略，進行資料同步作業，將下載同步資料存放在 Database Cache 內，作為緩衝資料使用。由於 Remote Service 在同步過程中一切順利，Remote Service 認定目前網路一切正常。同時間，SaaS 服務開始進行運

作醫療服務作業，當需要進行資料存取時，首先會向 Service Router 提出詢問資料存取路由請求，Service Router 會先確認網路狀況；經由 Remote Service 的回應，得知目前網路正常後，Service Router 會立即回應"存取 Remote Service" 訊息給 SaaS 服務，SaaS 服務收到回應後，透過 Remote Service 協同執行 Cloud Service 所提供的資料存取服務，此階段各元件間運作順序及關聯如圖 5 所示。

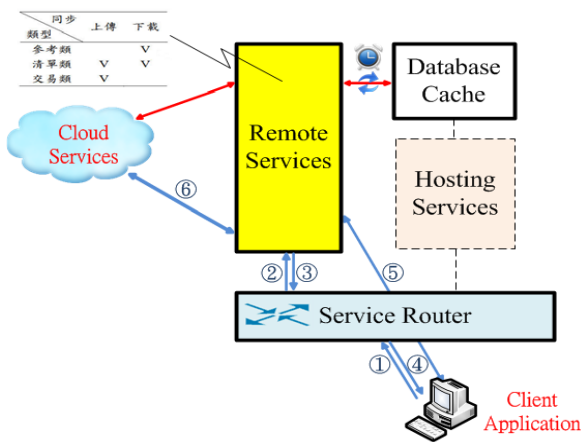


圖 5 永續運作之 SaaS 服務模式運作順序關聯圖 (網路服務正常運作時)

- 網路異常中斷時的 SaaS 服務流程：在 Remote Service 進行同步或協同 SaaS 服務執行 Cloud Service 作業時，發生了異常狀況，Remote Service 經過研判，確認目前網路異常(雖然已確認網路處於異常，Remote Service 還是持續監控網路狀況)。當 SaaS 服務再度向 Service Router 提出詢問資料存取路由請求時，Service Router 經由 Remote Service 的回應，得知目前網路異常後，Service Router 會立即回應"存取 Hosting Service" 訊息給 SaaS 服務，SaaS 服務收到回應後，即對 Hosting Service 提出資料存取請求，Hosting Service 開始依照 SaaS 服務請求，存取先前已同步過的 Database Cache 緩衝資料，此階段各元件間運作順序及關聯如圖 6 所示。

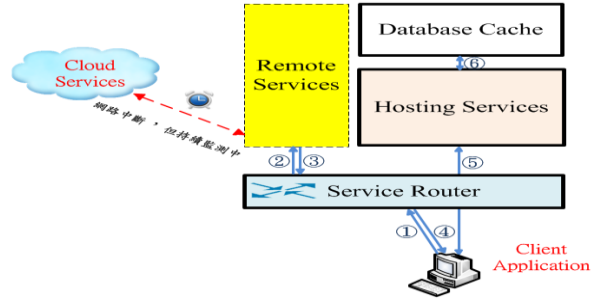


圖 6 永續運作之 SaaS 服務模式運作順序關聯圖 (網路服務中斷時)

- 網路服務由中斷回復成正常時的 SaaS 服務流程：當 Remote Service 持續監控網路過程，發現網路已恢復正常運作時，Remote Service 第一優先進行同步作業，將在網路中斷時，發生在用戶端 Database Cache 的變動資料，同步上傳回遠(雲)端資料庫中，此階段各元件間運作順序及關聯如圖 7 所示。待完成同步作業後，再回復至網路正常時運作機制。

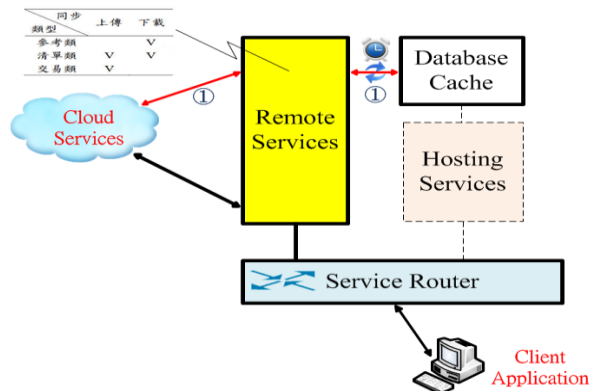


圖 7 永續運作之 SaaS 服務模式運作順序關聯圖 (網路恢復正常優先流程)

5. 結論

本研究係為了解決目前 SaaS 服務的缺陷，提出永續運作之 SaaS 服務模式，讓醫療診所在不增加軟硬體及人事開銷狀況下，得以享受使用 SaaS 服務帶來的各種優點，降低營運資訊成本，專注醫療本業，進而提昇醫療服務品質。

對於 SaaS 服務供應商來說，本研究架

構是建構在原本醫療資訊服務下，只需額外增加於用戶端安裝之 Sync Agenda 永續服務核心程式，即可達到永續服務之目的。在撰寫 Sync Agenda 程式時，擴充原有服務存取路由選擇功能，並利用 Sync Framework 技術，配合研究提出之資料變動同步策略矩陣，在同樣醫療商業邏輯下，提供額外存取暫存資料之 Cloud Services，即可跳脫原本服務框架限制，達到在網路不穩定或斷線時，提供永續醫療服務目的。另外，本研究完全遵循 SOA 架構，預留服務無限擴充能力，能夠快速滿足用戶因系統整合，衍生額外資料存取之需求。

參考文獻

- [1] 王耀聰，*淺談雲端運算的新趨勢*，元智大學資工系 Seminar，桃園縣，2010。
- [2] 石俊榮，*衛生單位對資訊安全系統執行現況之調查*，嘉南藥理科技大學碩士論文，台南市，2010。
- [3] 肖承勇，*設計偶爾連接的智能客戶端*，西南交通大學碩士論文，四川，2007。
- [4] 陳澄 等著，*雲端策略-雲端運算與虛擬化技術*，天下雜誌，台北市，2010。
- [5] 資策會產業情報研究所，*全球千億美元雲端商機-資策會打造雲端智慧：「整合雲與端的服務平台」*，資訊工業策進會，台北市，2010。
- [6] 葉偉 等著，*互聯網時代的軟件革命：SaaS 架構設計*，電子工業出版社，北京，2009。
- [7] 蔡承宏、吳文智、鄭涵綾、蔡榮隆，*醫院服務雲端平台之設計 - 以 RFID 應用系統為例*，台灣國際醫學資訊聯合研討會，台灣醫學資訊學會，2010。
- [8] 蕭惠如、王振彥、王思雁、李冠憬、溫嘉憲，*架構依需取用臨床資訊服務之 SaaS 雲端平台*，台灣國際醫學資訊聯合研討會，台灣醫學資訊學會，2010。
- [9] gipi，*三種雲端服務*。取自 gipi 的學習筆記-職場觀念、管理、IT 部落格：
<http://www.dotblogs.com.tw/jimmyyu/archive/2009/12/03/12275.aspx>，2010。
- [10] Anthony T. Velte. Toby Velte. Robert Elsenpeter, *Cloud Computing, A Practical Approach*, McGraw-Hill, NY, 2010.
- [11] Charles Babcock, *Management Strategies for the Cloud Revolution*, McGraw Hill, NY, 2010.
- [12] DRDO. Delhi. India, *Configurability in SaaS (software as a service) applications*, ISEC '09, 2009.
- [13] Edmund Tan, *NET Compact Framework 3.5 Data-Driven Applications*, Packt Publishing, Birmingham, UK, 2010.
- [14] Liang-Jie Zhang, Qun Zhou, *CCOA: Cloud Computing Open Architecture* IEEE Computer Society Washington, pp. 607-616, 2009.
- [15] John Rittinghouse. James Ransome, *Cloud Computing Implementation, Management and Security*, CRC Press, FL, 2010.
- [16] Microsoft Corporation. (n.d.) , *Introduction to Sync Framework Database Synchronization*, Retrieved from <http://msdn.microsoft.com/en-us/sync/bb887608.aspx>, 2010.
- [17] miller, *Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online*, Que Pub, NJ, 2008.
- [18] Nick Randolph. David Gardner. Chris Anderson. Michael Minutillo, *Professional Visual Studio 2010*, Addison-Wesley, MA, 2010.
- [19] Peter Mell. Tim Grance, *The NIST Definition of Cloud Computing*, Retrieved from <http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc> , 2010.
- [20] Sharp John, *Service Oriented Architectures and Windows Communication Foundation*, Microsoft Press, WA, 2007.